

An Isabelle Overview

Wenzel, Paulson, *Nipkow*, Krauss, Haftmann, Chaieb,
Berghofer, Ballarin, . . .

München & Cambridge

What is Isabelle?

A logical framework based on minimal higher-order logic

What is Isabelle?

A logical framework based on minimal higher-order logic

Main instances:

HOL

ZF

Topics

- Definitions and proofs in Isabelle/HOL —
The `Isar` language

Topics

- Definitions and proofs in Isabelle/HOL —
The `Isar` language
- Proof automation

Topics

- Definitions and proofs in Isabelle/HOL —
The `Isar` language
- Proof automation
- More ...

fun
datatype
inductive

Isar

fun

datatype

inductive

class Type classes

locale Named contexts

Proof Automation

FOL/HOL

Proof Automation

FOL/HOL

- `simp` & `auto`

Proof Automation

FOL/HOL

- `simp` & `auto`
- `sledgehammer`

Proof Automation

FOL/HOL

- `simp` & `auto`
- `sledgehammer`
- `metis` (Hurd)

Proof Automation

FOL/HOL

- `simp` & `auto`
- `sledgehammer`
- `metis` (Hurd)
- `blast`

Proof Automation

FOL/HOL

- `simp` & `auto`
- `sledgehammer`
- `metis` (Hurd)
- `blast`

Arithmetic `arith`

Proof Automation

FOL/HOL

- `simp` & `auto`
- `sledgehammer`
- `metis` (Hurd)
- `blast`

Arithmetic `arith`

Algebra `algebra` (Harrison/Buchberger/Gröbner)

More

- refute and quickcheck

More

- `refute` and `quickcheck`
- Code generation

More

- `refute` and `quickcheck`
- Code generation
- Other logics:

More

- `refute` and `quickcheck`
- Code generation
- Other logics:
 - `Nominal` (Urban)

More

- `refute` and `quickcheck`
- Code generation
- Other logics:
 - `Nominal` (Urban)
 - `ZF` (Kolodynski)

More

- [refute](#) and [quickcheck](#)
- Code generation
- Other logics:
 - [Nominal](#) (Urban)
 - [ZF](#) (Kolodynski)
- [Archive of Formal Proofs](#)

More

- `refute` and `quickcheck`
- Code generation
- Other logics:
 - `Nominal` (Urban)
 - `ZF` (Kolodynski)
- `Archive of Formal Proofs`
- `LATEX`

Sample Applications

- Programming language definitions:
Java and JVM dialect [Jinja](#) (Klein)

Sample Applications

- Programming language definitions:
Java and JVM dialect [Jinja](#) (Klein)
- Analysis and number theory:
[Prime Number Theorem](#) (Avigad)

Sample Applications

- Programming language definitions:
Java and JVM dialect [Jinja](#) (Klein)
- Analysis and number theory:
[Prime Number Theorem](#) (Avigad)
- Hales' proof of [Kepler Conjecture](#) (Bauer, Obua)

Sample Applications

- Programming language definitions:
Java and JVM dialect [Jinja](#) (Klein)
- Analysis and number theory:
[Prime Number Theorem](#) (Avigad)
- Hales' proof of [Kepler Conjecture](#) (Bauer, Obua)
- System Verification:
[VeriSoft](#) (Paul) and [L4.verified](#) (Klein)

Conclusion

Conclusion

Hope to see you all at TPHOLs 2009 in Munich!

Conclusion

Hope to see you all at TPHOLs 2009 in Munich!



Conclusion

Hope to see you all at TPHOLs 2009 in Munich!



Conclusion

Hope to see you all at TPHOLs 2009 in Munich!

